Riešenie konfliktov pri LR analýze, GLR analyzátor

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

viliam.hromada@stuba.sk

Riešenie konfliktov pri syntaktickej analýze bottom-up

- Ani použitie pokročilejších analyzátorov typu *LALR*(1) nezaručí, že každá bezkontextová gramatika sa dá deterministicky parsovať.
- V praxi sa často vyskytujú nejednoznačné gramatiky, ktoré sa už z definície nedajú parsovať učebnicovým použitím *SLR*(1) či *LALR*(1), - pretože obsahujú konflikt.
- Avšak, aj tak sa **prakticky použijú**, len s tým rozdielom, že v prípade konfliktu je jednoznačne určené, ktorá akcia sa má vykonať - určí sa **priorita** operácií, t.j. či sa v prípade konfliktu **presun-redukcia** vykoná **presun** alebo **redukcia**, resp. v prípade **konfliktných redukcií** sa povie, **ktorá redukcia** sa má vykonať.

Riešenie konfliktov pri `if-then-else` probléme

- Vráťme sa ku zjednodušenej gramatike vetvenia:
  1. <príkaz> → `if`<podmienka>`then`<príkaz><else časť>
  2. <príkaz> → `iný`
  3. <else časť> → `else`<príkaz>
  4. <else časť> → ε
  5. <podmienka> → `p`.

Riešenie konfliktov pri `if-then-else` probléme

- Jeden zo stavov *LALR*(1)-automatu bude obsahovať nasledovné položky:
  1. <príkaz> → `if`<podmienka>`then`<príkaz> • <else časť>, $\{\varepsilon, \texttt{else}\}$
  2. <else časť> → •`else`<príkaz>, $\{\varepsilon, \texttt{else}\}$
  3. <else časť> → •, $\{\varepsilon, \texttt{else}\}$
- Teda je tam konflikt presun/redukcia pre vstupný symbol `else`, ktorý je spôsobený tým, že gramatika **nie je jednoznačná**.
- Ak by sme si povedali, že `else` sa vždy viaže s najbližším predchádzajúcim `if`-om, tak v takom prípade uprednostníme Presun pred redukciou.

Riešenie konfliktov pri syntaktickej analýze bottom-up - *GLR*

- Iná metóda, ako priorita, je tzv. GLR-parser (Generalized LR parser).
- Dá sa zostrojiť z hocakého *LR*-analyzátora.
- *GLR* syntaktický analyzátor pre svoju činnosť používa taktiež tabuľky *ACTION* a *GOTO*, avšak neprekáža, ak je v tabuľke *ACTION* konflikt.
- V prípade, že $ACTION[s, t]$ pripúšťa viacero akcií, tak sa výpočet **rozvetví** na viacero procesov, čím sa v podstate simuluje prechádzanie **všetkých možností** výpočtu analyzátora.
- Prakticky sa naň dá nazerať ako na nedeterministický syntaktický analyzátor (zásobníkový automat).

GLR analyzátor

- V momente, keď sa výpočet vetví, vzniká nová vetva (nový proces) s vlastným zásobníkom.
- Vstupné symboly sa spracúvajú paralelne vo všetkých procesoch - na nový symbol sa prejde až potom, ako všetky aktuálne procesy spracujú aktuálny vstupný symbol.
- Ak niektorý proces narazí na syntaktickú chybu, tak sa ukončí.
- Ak aspoň jeden z procesov skončí akceptáciou, tak sa vstupné slovo akceptuje (t.j. existuje akceptačný výpočet).

Príklad

Nájdite odvodenie slova *cda* pomocou *GLR* analyzátora v gramatike

$S \to A$

$A \to BDa$

$A \to CDb$

$B \to \varepsilon$

$C \to \varepsilon$

$D \to cd$

Príklad - príslušný *LALR*(1)-automat

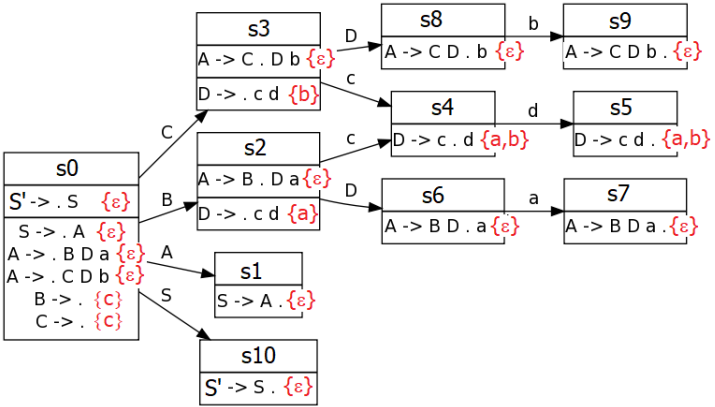

Figure: Prevzaté z Dedera, Ľ.: Počítačové jazyky a ich spracovanie

Príklad - príslušná *ACTION* tabuľka *SLR*(1)-analyzátora

ACTION	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
a						R6	P				
b						R6			P		
c	R4,R5		P	P							
d					P						
ε		R1						R2		R3	A

Príklad - výpočet *GLR* analyzátora

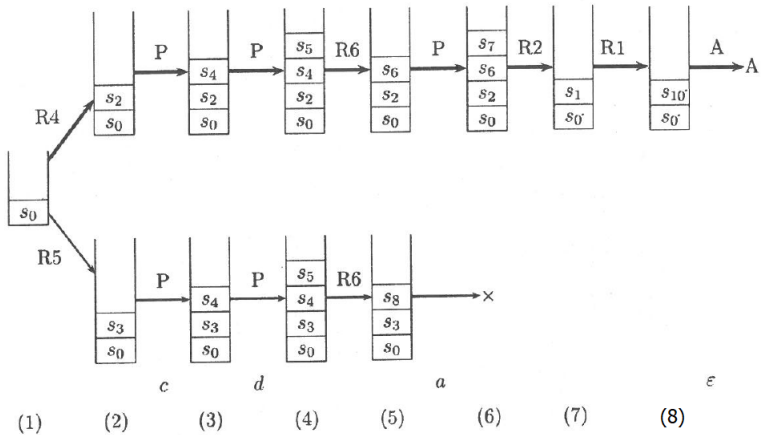

Figure: Prevzaté z Dedera, Ľ.: Počítačové jazyky a ich spracovanie

Optimalizácia *GLR*-analyzátora

- Bezprostredne po rozvetvení sú obsahy zásobníkov rovnaké - preto použijeme reprezentáciu, kde bude rovnaká **história zdieľaná** medzi zásobníkmi.
- Ak sa na vrchu zásobníkov procesov zjaví rovnaký stav a bude sa konať presun, môžu sa zlúčiť, pretože sa ďalej budú vyvíjať zhodne.
- Ak nejaká redukcia odstráni celú zlúčenú časť zásobníka, je potrebné obnoviť aj pôvodné zásobníky.
- Používa sa preto tzv. *zásobník s grafovou štruktúrou* - Graph Structured Stack.

Príklad - výpočet *GLR* analyzátora

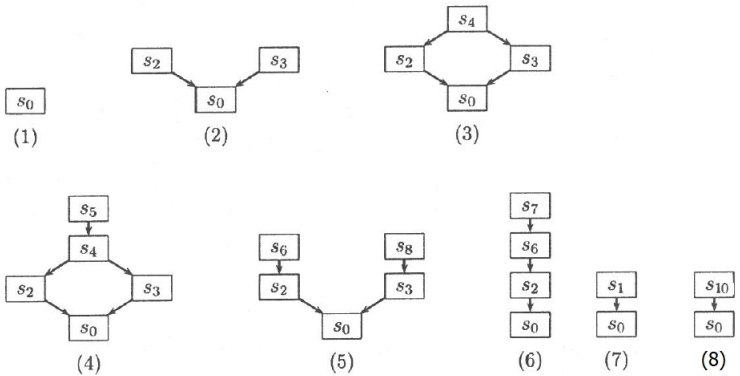

Figure: Prevzaté z Dedera, Ľ.: Počítačové jazyky a ich spracovanie

Vzťahy medzi bottom-up parsermi

- Každá *LR*(0) gramatika je *SLR*(1) gramatika.
- Každá *SLR*(1) gramatika je *LALR*(1) gramatika.

Vzťahy medzi *LL* a *LR*

1. Ak G je $LL(1)$-gramatika bez ε-pravidiel, potom je aj $LR(0)$-gramatikou.
2. Niektoré triedy nie sú porovnateľné, napr. $LL(1)$ gramatiky a $LALR(1)$ gramatiky - existujú gramatiky $LL(1)$, ktoré nie sú $LALR(1)$ a naopak.

Použitá literatúra

Aho, A., Lam, M., Sethi, R., Ullman, J.: *Compilers: Principles, techniques and tools.*

Dedera, Ľ.: *Počítačové jazyky a ich spracovanie*.

Linz, P.: *An Introduction to Formal Languages and Automata*.